



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/698,820	10/31/2003	Matthew Englehart	MWS-062	1288
959	7590	10/20/2006	EXAMINER	
LAHIVE & COCKFIELD, LLP ONE POST OFFICE SQUARE BOSTON, MA 02109-2127			CHEN, QING	
		ART UNIT	PAPER NUMBER	
		2191		

DATE MAILED: 10/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/698,820	ENGLEHART ET AL.	
	<b>Examiner</b>	<b>Art Unit</b>	
	Qing Chen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) Responsive to communication(s) filed on 31 October 2003.
- 2a) This action is FINAL.                    2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G./213.

#### Disposition of Claims

- 4) Claim(s) 1-26 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_\_ is/are allowed.
- 6) Claim(s) 1-26 is/are rejected.
- 7) Claim(s) \_\_\_\_\_ is/are objected to.
- 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 31 October 2003 is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All    b) Some \* c) None of:
1. Certified copies of the priority documents have been received.
  2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)          | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____                                      |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____.   | 6) <input type="checkbox"/> Other: _____.                         |

## DETAILED ACTION

1. This is the initial Office action based on the application filed on October 31, 2003.

**Claims 1-26** are currently pending and have been considered below.

### *Specification*

2. The disclosure is objected to because of the following informalities:

- The specification contains the following typographical errors:
  - In the “Related Applications” section, the patent application number should be designated as “United States Patent Application No.”
  - There is a duplicated period (.) at the end of the paragraph on page 2, line 2.
  - The word “and” should be added after the description for Figure 3 in page 3.
  - A period (.) should be added after the description for Figure 4 in page 3.
  - The specification contains numerous instances of the terminology “custom class.” It should presumably read -- custom storage class --.
  - A white space should be added between the number “12” and the phrase “(step 62)” in page 8.
  - Reference numbers 84, 86, 88, 90, 92, and 94 are used to designate both user interface parameter settings and types of user interface presentation elements in pages 8 and 9. For example, reference number 84 is used to designate both “visibility parameter setting” and “pull down menu.”
  - The word “by” in the phrase “... selected parameter setting may be implemented without *by* the automatic code generator” should be deleted in page 9.

Appropriate correction is required.

***Claim Objections***

3. **Claims 2-10, 12-15, and 17-25** are objected to because of the following informalities:

- **Claims 2, 3, and 18** recite the limitation “the code.” The Examiner subsequently interprets this limitation as reading “the source code” for the purpose of providing it with proper explicit antecedent basis.
- **Claims 3 and 18** contain a typographical error: the word “step” in the preamble should be pluralized, since the claims are further limited with multiple steps.
- **Claims 3, 13, and 18** recite the limitation “said custom class.” The Examiner subsequently interprets this limitation as reading “said custom storage class” for the purpose of providing it with proper explicit antecedent basis.
- **Claims 3, 13, and 18** contain a typographical error: a hyphen (-) should be added between the words “user” and “selected.” Applicant is advised to make the correction in order to keep the terminology consistent throughout the specification and claims.
- **Claims 4, 14, and 19** recite the limitation “the adjusted code.” The Examiner subsequently interprets this limitation as reading “the adjusted source code” for the purpose of providing it with proper explicit antecedent basis.
- **Claims 4, 14, and 19** depend on Claims 3, 13, and 18, respectively. Therefore, these claims suffer the same deficiencies as their respective parent claims.

- **Claims 4, 5, 14, 15, 19, and 20** recite the limitation “said view of code.” The Examiner subsequently interprets this limitation as reading “said view of salient aspects of the source code” for the purpose of providing it with proper explicit antecedent basis.
- **Claim 5** depends on Claim 2 and, therefore, suffers the same deficiency as Claim 2.
- **Claims 5 and 20** recite the limitation “segment of code.” The Examiner subsequently interprets this limitation as reading “segment of source code” for the purpose of providing it with proper explicit antecedent basis.
- **Claims 5-10, 13-15, and 20-25** contain a typographical error: a comma (,) should be added between the parent claim number and the word “wherein.”
- **Claims 8 and 23** recite the limitation “automatically generated code.” The Examiner subsequently interprets this limitation as reading “automatically generated source code” for the purpose of providing it with proper explicit antecedent basis.
- **Claim 12** contains a typographical error: the white space before the semicolon (;) at the end of the first limitation should be deleted.
- **Claims 12 and 17** recite the limitation “code.” The Examiner subsequently interprets this limitation as reading “the source code” for the purpose of providing it with proper explicit antecedent basis.
- **Claims 13-15 and 18-20** depend on Claims 12 and 17, respectively. Therefore, these claims suffer the same deficiency as their respective parent claims.

Appropriate correction is required.

***Claim Rejections - 35 USC § 101***

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. **Claims 12-26** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

**Claims 12-15** are directed to systems of functional descriptive material *per se*, and hence non-statutory. There are no indications or suggestions in the specification or claims that would associate the recited software components in the claims with hardware elements of the electronic device.

**Claims 16-26** recite a medium as a claimed element. However, the specification does not provide an explicit definition for what constitutes a medium. Thus, the claims can be reasonably interpreted as functional descriptive material *per se*, since the medium is not claimed as being computer-readable to provide the necessary functional and structural interrelationship to satisfy the requirements of 35 U.S.C. § 101. It is noted that the preamble of Claim 16 also recites that the medium holds executable steps. Although the steps of the methods are described as being executable, this does not associate the medium as being executable because there is no explicit indication that the medium can be executed in the claims.

Therefore, these claims constitute computer programs representing computer listings *per se*. Such descriptions or expressions of the programs are not physical “things.” They are neither computer components nor statutory processes, as they are not “acts” being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program’s functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element, which defines structural and functional interrelationships between the computer program and the rest of the computer, that permits the computer program’s functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

***Claim Rejections - 35 USC § 102***

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. **Claims 1-4, 6-8, 12-14, 16-19, and 21-23** are rejected under 35 U.S.C. 102(b) as being anticipated by Cheng et al. (US 2002/0010908).

As per **Claim 1**, Cheng et al. disclose:

Art Unit: 2191

- providing a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment (*see Figure 6: 330, 350, and 370; Paragraphs [0026], [0028], and [0043]*, “*The entering of parameters is also accomplished via GUI by adding the desired parameters to parameter field*” and “*... the command structure is generated by command structure generation engine using the information entered by the developer*” and “*... the handler function code is generated by handler code generation engine*”); and
- creating a custom storage class in said graphical modeling and execution environment utilizing parameters selected by a user from said plurality of selectable parameters (*see Figure 6: 360; Paragraph [0039]*, “*... the handler function definitions and parameter definitions are generated by command structure generation engine*” and “*command structure generation engine takes the information input by the developer and generates a file containing the information for the handler functions and parameters*”).

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Cheng et al. further disclose:

- providing a view of salient aspects of the source code generated by said automatic code generator utilizing the user-selected parameters (*see Figure 11; Paragraph [0046]*, “*GUI displays the code generated by handler code generation engine so that the developer may view, review and accept the automatically generated code*”).

As per **Claim 3**, the rejection of **Claim 2** is incorporated; and Cheng et al. further disclose:

- changing the user-selected parameters for said custom storage class in said user interface (*see Paragraph [0051]*, “*... a developer edits parameters in a handler function through GUI ...*”); and
  - adjusting the source code generated by said automatic code generator to reflect the change in user-selected parameters (*see Paragraph [0051]*, “*the command structure, the handler function definitions, the parameter definitions and the handler function code is automatically generated based on the information provided by the developer and therefore may need to be revised based on any changed or additional information provided by the developer*” and “*... these changes will be automatically reconciled in the handler function code by handler code generation engine*”).

As per **Claim 4**, the rejection of **Claim 3** is incorporated; and Cheng et al. further disclose:

- displaying salient aspects of the adjusted source code in said view of salient aspects of the source code (*see Paragraph [0044]*, “*This code may be viewed as it is being generated in code view field of GUI as parameters are being added to the handler function*”).

As per **Claim 6**, the rejection of **Claim 1** is incorporated; and Cheng et al. further disclose:

Art Unit: 2191

- wherein said custom storage class declares macros for instances of constant data (see Paragraph [0038], "... #define kCommand3Help "\help string for command 3\" ...").

As per **Claim 7**, the rejection of **Claim 1** is incorporated; and Cheng et al. further disclose:

- wherein said custom storage class declares variables for instances of constant data (see Paragraph [0040], "The exemplary code may include the following information: keyword or name, data type (e.g., integer, boolean, etc.), a unique bitmask identifier ...").

As per **Claim 8**, the rejection of **Claim 1** is incorporated; and Cheng et al. further disclose:

- wherein said user-selected parameters control at least one of the manner in which automatically generated source code is defined, declared, accessed and addressed (see Paragraph [0043], "Handler code generation engine automatically generates this software code using the information entered by the developer and the parameter and handler function definitions generated by command structure generation engine").

As per **Claim 12**, Cheng et al. disclose:

- a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code from said graphical model (see Figure 6: 330, 350, and 370; Paragraphs [0026], [0028], and [0043], "The entering of parameters is also accomplished via GUI by adding the

*desired parameters to parameter field” and “... the command structure is generated by command structure generation engine using the information entered by the developer” and “... the handler function code is generated by handler code generation engine”);*

- a custom storage class in said graphical modeling and execution environment, said custom storage class created utilizing parameters selected by a user from said plurality of selectable parameters (*see Figure 6: 360; Paragraph [0039], “... the handler function definitions and parameter definitions are generated by command structure generation engine” and “command structure generation engine takes the information input by the developer and generates a file containing the information for the handler functions and parameters”*); and
- a view of salient aspects of the source code generated by said automatic code generator utilizing the user-selected parameters (*see Figure 11; Paragraph [0046], “GUI displays the code generated by handler code generation engine so that the developer may view, review and accept the automatically generated code”*).

As per **Claim 13**, the rejection of **Claim 12** is incorporated; and Cheng et al. further disclose:

- wherein the user-selected parameters for said custom storage class in said user interface are changed and the source code generated by said automatic code generator is adjusted to reflect the change user-selected parameters (*see Paragraph [0051], “... a developer edits parameters in a handler function through GUI ...” and “the command structure, the handler function definitions, the parameter definitions and the handler function code is automatically generated based on the information provided by the developer and therefore may need to be*

*revised based on any changed or additional information provided by the developer” and “...  
these changes will be automatically reconciled in the handler function code by handler code  
generation engine”).*

As per **Claim 14**, the rejection of **Claim 13** is incorporated; and Cheng et al. further disclose:

- wherein the adjusted source code is displayed in said view of salient aspects of the source code (*see Paragraph [0044], “This code may be viewed as it is being generated in code view field of GUI as parameters are being added to the handler function”*).

**Claims 16-19 and 21-23** are medium claims corresponding to the method claims above (Claims 1-4 and 6-8, respectively) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1-4 and 6-8, respectively.

#### ***Claim Rejections - 35 USC § 103***

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. **Claims 5, 15, and 20** are rejected under 35 U.S.C. 103(a) as being unpatentable over Cheng et al. (US 2002/0010908) in view of Childress et al. (US 2004/0085357).

As per **Claim 5**, the rejection of **Claim 2** is incorporated; however, Cheng et al. do not disclose:

- wherein said view of salient aspects of the source code automatically generated includes at least one token, said token being symbolically representative of a non-displayed segment of source code.

Childress et al. disclose wherein said view of salient aspects of the source code automatically generated includes at least one token, said token being symbolically representative of a non-displayed segment of source code (*see Paragraph [0115]*, “*... code may be included as ‘hidden’ text in one or more sections of documents, and may be used in constructing header tables and text tables*”).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Childress et al. into the teaching of Cheng et al. to include wherein said view of salient aspects of the source code automatically generated includes at least one token, said token being symbolically representative of a non-displayed segment of source code. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize the usage of available memory.

As per **Claim 15**, the rejection of **Claim 12** is incorporated; however, Cheng et al. do not disclose:

- wherein said view of salient aspects of the source code includes at least one token, said token being symbolically representative of a non-displayed segment of code.

Childress et al. disclose wherein said view of salient aspects of the source code includes at least one token, said token being symbolically representative of a non-displayed segment of code (*see Paragraph [0115]*, “*... code may be included as ‘hidden’ text in one or more sections of documents, and may be used in constructing header tables and text tables*”).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Childress et al. into the teaching of Cheng et al. to include wherein said view of salient aspects of the source code includes at least one token, said token being symbolically representative of a non-displayed segment of code. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize the usage of available memory.

**Claim 20** is rejected for the same reason set forth in the rejection of Claim 5.

10. **Claims 9, 10, 24, and 25** are rejected under 35 U.S.C. 103(a) as being unpatentable over Cheng et al. (US 2002/0010908) in view of Davidov et al. (US 2003/0225774).

As per **Claim 9**, the rejection of **Claim 1** is incorporated; however, Cheng et al. do not disclose:

- wherein said user-selected parameter includes a non-portable directive to a compiler.

Davidov et al. disclose wherein said user-selected parameter includes a non-portable directive to a compiler (*see Paragraphs [0092] and [0214]*, “*Command line options or*

*directives for the compiler ...” and “The compiler compiles Java source code produced by the generator according to supplied directives”).*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Davidov et al. into the teaching of Cheng et al. to include wherein said user-selected parameter includes a non-portable directive to a compiler. The modification would be obvious because one of ordinary skill in the art would be motivated to conveniently and dynamically create software programs that can be executed on a computer system.

As per **Claim 10**, the rejection of **Claim 9** is incorporated; however, Cheng et al. do not disclose:

- wherein said non-portable directive to a compiler assigns data to at least one memory location in said electronic device.

Davidov et al. disclose wherein said non-portable directive to a compiler assigns data to at least one memory location in said electronic device (*see Paragraph [0160], “The data is loaded when the application is started, and is saved when the application is destroyed. This type of persistence uses the device records management system (RMS), for example, non-volatile memory”*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Davidov et al. into the teaching of Cheng et al. to include wherein said non-portable directive to a compiler assigns data to at least one memory

location in said electronic device. The modification would be obvious because one of ordinary skill in the art would be motivated to store data that can be utilized at a later time.

**Claim 24** is rejected for the same reason set forth in the rejection of Claim 9.

**Claim 25** is rejected for the same reason set forth in the rejection of Claim 10.

11. **Claims 11 and 26** are rejected under 35 U.S.C. 103(a) as being unpatentable over Cheng et al. (US 2002/0010908) in view of DeMaster (US 6,066,181).

As per **Claim 11**, the rejection of **Claim 1** is incorporated; however, Cheng et al. do not disclose:

- creating a separate header file with said automatic code generator in response to the selection of one of said plurality of user-selected parameters.

DeMaster discloses creating a separate header file with said automatic code generator in response to the selection of one of said plurality of user-selected parameters (*see Column 4: 55-57, "... the Java native interface code generator generates Java Classes and data conversion code stubs (and related header files)"*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of DeMaster into the teaching of Cheng et al. to include creating a separate header file with said automatic code generator in response to the selection of one of said plurality of user-selected parameters. The modification would be obvious because one of ordinary skill in the art would be motivated to allow software portability, so that

software applications may easily be moved to another environment (*see DeMaster – Column 1: 23-25*).

**Claim 26** is rejected for the same reason set forth in the rejection of Claim 11.

### *Conclusion*

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- A. Marmelstein (US 5,187,788) discloses a graphics system and method for generation of code in the Ada language.
- B. Lau et al. (US 5,892,510) disclose a structure and method for facilitating development of graphical user interfaces for databases.
- C. McDonald et al. (US 5,966,532) disclose a system and method for automatically generating graphical program code based on user interface configurations created by the user.
- D. Pavela (US 6,421,822) discloses a method and system for generating test cases using a test object library.
- E. Bailey et al. (US 6,684,385) disclose a program-development environment that allows developers to seamlessly switch between a visual programming paradigm and a textual programming paradigm, thereby permitting developers to choose the paradigm best suited for creating each aspect of the desired program.

F. Hunt (US 2003/0056195) discloses a computer program, method and system for generating computer code to program objects, interfaces, factories and their relational schema using languages such as C++ and SQL.

G. Gregory (US 2003/0074235) discloses a computer program responsive to a user defined and built database including elements of the database and unique operating rules all used to generate an individual database and executable code.

H. Soroker et al. (US 2004/0111696) disclose a method and system for an application builder that can optionally generate code used for debugging or testing purposes.

I. Bryant et al. (US 2004/0230945) disclose invocation of code-generation functionality in a graphical programming language, e.g., via integration of a configuration tool with the programming language.

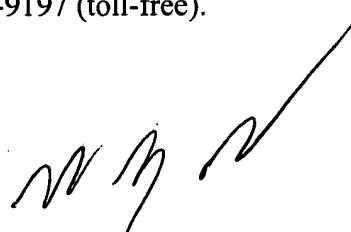
J. Santori et al. (US 2004/0255269) disclose a system and method for creating an association between a graphical user interface element and a graphical program, e.g., a block diagram, wherein the GUI element can provide data to and/or receive data from the graphical program.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

Art Unit: 2191

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



WEI ZHEN  
SUPERVISORY PATENT EXAMINER

QC / QC  
October 16, 2006